Enhanced Convergent PNP Algorithms For Image Restoration

ENHANCED CONVERGENT PNP ALGORITHMS FOR IMAGE RESTORATION

Matthieu Terris[⋆⋆]*, Audrey Repetti*[⋆◇]*, Jean-Christophe Pesquet*[‡†]*, and Yves Wiaux*[⋆]

[⋆] Institute of Sensors, Signals and Systems, Heriot-Watt University, Edinburgh EH14 4AS, United Kingdom

[◇] Department of Actuarial Mathematics & Statistics, Heriot-Watt University, Edinburgh EH14 4AS, United Kingdom

[‡] Université Paris-Saclay, CentraleSupélec, Inria, Center for Visual Computing, 91190 Gif sur Yvette, France

ABSTRACT

Image restoration has long been one of the key research topics in image processing. Many mathematical approaches have been developed to solve this problem, e.g., variational methods, wavelet techniques, or Bayesian methods. With the widespread of neural network (NN) models in all the subdomains of data science, the performance limits of these methods are further pushed. One of the most successful strategies consists of plugging NNs in existing optimization algorithms. However, so doing raises several mathematical and practical challenges. One of the main issues is to secure the convergence of the resulting iterative scheme. Further questions concerning the characterization of the reached limit are also worth being addressed. In this paper, we show that the theory of maximally monotone operators allows us to bring insightful answers to these problems and to design firmly nonexpansive NNs; combining these with postprocessing NNs leads to excellent global restoration quality.

Index Terms— Neural networks, forward-backward algorithm, plug-and-play algorithm, image restoration

1. INTRODUCTION

Image restoration problems can often be formulated as inverse problems where the objective is to recover an original unknown image $\overline{x}$, assumed to belong to a real Hilbert space $\mathcal{H}$, from degraded measurements $z \in \mathcal{H}$ given by

$$z = H\overline{x} + e, \tag{1}$$

with $H\colon \mathcal{H} \to \mathcal{H}$ a degradation linear operator and $e \in \mathcal{H}$ the realization of an additive random noise. A standard way of solving (1) is to define the estimate $\widehat{x} \in \mathcal{H}$ of $\overline{x}$ as a minimizer of the sum of two functions: a data-fidelity term $f \in \Gamma_0(\mathcal{H})$[1] related to the noise distribution in (1), and a regularization term $g \in \Gamma_0(\mathcal{H})$ incorporating *a priori* information one has on the target image. For many standard noise distributions (e.g. Gaussian, Poisson-Gauss, or logistic), f corresponds to a μ-Lipschitz differentiable function. Under qualifications conditions, this minimization problem is equivalent to

$$\text{find } \widehat{x} \in \mathcal{H} \text{ such that } \quad 0 \in \nabla f(\widehat{x}) + \partial g(\widehat{x}). \tag{2}$$

This problem can be solved efficiently using proximal methods (see e.g., [1]). In particular, a standard approach consists in using a forward-backward (FB) algorithm [2, 3, 4]:

$$(\forall n \in \mathbb{N}) \qquad x_{n+1} = J_{\gamma \partial g}\left(x_n - \gamma \nabla f(x_n)\right), \tag{3}$$

where $0 < \gamma < 2/\mu$, and $J_{\gamma\partial g}$ is the resolvent of $\gamma\partial g$, corresponding to the proximity operator of g. The sequence $(x_n)_{n\in\mathbb{N}}$ is known to converge (weakly) to a solution to (2).

The regularization term g is often necessary to avoid ill-posedness problems, and many works have been dedicated to the choice of this term. It often tends to promote the smoothness of the solution or to enforce its sparsity. Good examples of such regularization functions are the total variation semi-norm [5] and its various extensions [6, 7], and sparsity-promoting functions leveraging wavelet decomposition [8]. Recently, Plug-and-Play (PnP) approaches [9] suggested to replace the proximity operators naturally arising in proximal algorithms used for solving (2) by a denoiser. This strategy has since then become the state-of-the-art in many applications [10, 11], in particular with the recent progress of learning-based denoisers and Neural Networks (NNs).

The theoretical convergence of sequences generated by PnP algorithms has drawn a lot of attention in the last years [12, 13, 14, 15]. However, it often relies on strong structural constraints on the denoiser (e.g. NNs without residual skip connection) [14, 15], and/or the limit point is not clearly characterized [12, 13].

In this work, we employ a technique to train any NN as the resolvent of a maximally monotone operator, ensuring the convergence of the resulting PnP algorithm to a solution to a variational inclusion problem. Leveraging this technique, we develop a convergent PnP-FB algorithm, paired with a post-processing approach. We show that the resulting enhanced

[⋆⋆]This work was partly supported by the UK Engineering and Physical Sciences Research Council (EPM008843/1 and EP/M019306/1) and used the Cirrus UK National Tier-2 HPC Service at EPCC (http://www.cirrus.ac.uk) funded by the University of Edinburgh and EPSRC (EP/P020267/1).

[†]This work was partly supported by Institut Universitaire de France and the ARN Chair BRIDGEABLE in AI.

[1] $\Gamma_0(\mathcal{H})$ denotes the set of convex, proper and lower-semicontinuous function from $\mathcal{H}$ to $]-\infty, +\infty]$.

PnP method leads to very competitive results compared to the most advanced (but heuristic) NN-based image restoration methods.

The paper is organized as follows: our approach is given in Section 2. Then, in Section 3 we show the high quality of the results achieved with our method in image deconvolution problems. Finally, we draw some brief conclusions.

2. PROPOSED METHOD

In this section we detail the proposed method, in three steps. We first define the PnP-FB algorithm, providing convergence guaranties. We then give the proposed training scheme to ensure the convergence of the PnP-FB algorithm. Finally, we develop a post-processing technique to improve the final reconstruction results.

2.1. Convergent PnP algorithms

To build a PnP algorithm based on the FB iteration (3), we need to replace the resolvent operator $J_{\gamma\partial g}$ by a denoising operator $\widetilde{J}\colon \mathcal{H} \to \mathcal{H}$, leading to the following PnP-FB algorithm:

$$(\forall n \in \mathbb{N}) \qquad x_{n+1} = \widetilde{J}\left(x_n - \gamma \nabla f(x_n)\right). \tag{4}$$

It is well known from fixed point theory that a sufficient condition for (4) to converge is to have $\widetilde{J}$ firmly nonexpansive [4]. Recall that the operator $\widetilde{J}$ is firmly nonexpansive if, for every $(x, y) \in \mathcal{H}^2$, $\|\widetilde{J}x - \widetilde{J}y\|^2 \leqslant \left\langle x - y \mid \widetilde{J}x - \widetilde{J}y \right\rangle$.

To build such operator $\widetilde{J}$, we can leverage the following classical result from monotone operator theory [4].

Proposition 2.1. *Let $A\colon \mathcal{H} \rightrightarrows \mathcal{H}$. A is a maximally monotone operator if and only if its resolvent is firmly nonexpansive, i.e. there exists a nonexpansive (i.e. 1-Lipschitzian) operator $Q\colon \mathcal{H} \to \mathcal{H}$ such that*

$$J_A\colon \mathcal{H} \to \mathcal{H}\colon x \mapsto \frac{x + Q(x)}{2}. \tag{5}$$

In turn, $A = 2(\mathrm{Id} + Q)^{-1} - \mathrm{Id}$.

This proposition has two important consequences. First, according to (5), to build a firmly nonexpansive operaor $\widetilde{J}$, one can build a 1-Lipschitzian operator $Q = (2\widetilde{J} - \mathrm{Id})$ instead and then deduce $\widetilde{J}$. Second, if $\widetilde{J}$ is firmly nonexpansive, then there exists a maximally monotone operator A such that $\widetilde{J} = J_A$. Using this second statement, we can deduce the following proposition, the proof of which is given in [16], stating the convergence of sequences $(x_n)_{n\in\mathbb{N}}$ generated by the PnP-FB algorithm (4).

Proposition 2.2. *Let $\mu \in \left]0, +\infty\right[$ and let $\gamma \in \left]0, 2/\mu\right[$. Let $f\colon \mathcal{H} \to \mathbb{R}$ be a convex differentiable function with μ-Lipschitzian gradient. Let $\widetilde{J}$ be a neural network such that $\widetilde{J}$ is firmly nonexpansive as in* (5)*. Let $\widetilde{A}$ be the maximally monotone operator equal to $(\widetilde{J}^{-1} - \mathrm{Id})$. Assume that the set $\mathcal{S}_\gamma$ of zeros of $\nabla f + \gamma^{-1}\widetilde{A}$ is nonempty. Then, the sequence $(x_n)_{n\in\mathbb{N}}$ generated by iteration* (4) *converges (weakly) to $\widehat{x} \in \mathcal{S}_\gamma$, i.e., $\widehat{x}$ satisfies*

$$0 \in \nabla f(\widehat{x}) + \gamma^{-1}\widetilde{A}(\widehat{x}). \tag{6}$$

Conversely, if $\widehat{x}$ is a solution to (6) *and $\widetilde{A}$ is an arbitrary stationary maximally monotone operator, then its resolvent can be approximated as closely as desired by some firmly nonexpansive neural network.*

Three remarks can be made on Proposition 2.2. First, the variational inclusion (6) is more general than the one arising from the traditional Bayesian formulation (2). Indeed, unlike (2), (6) does not necessarily correspond to a minimization problem. Second, the variational inclusion (2) depends on the stepsize γ. This has an interesting echo in the PnP literature where the dependency of the solution to the algorithm parameters has long been acknowledged [17, 18]. Last, the final part of the proposition constitutes a universal approximation result showing that neural networks can provide good approximations to the resolvents of a wide class of maximally monotone operators.

2.2. Training firmly nonexpansive networks

We propose to train $\widetilde{J}$ as a denoiser and to ensure its firm nonexpansiveness.

Let $\overline{\boldsymbol{x}} = (\overline{x}_i)_{1\leqslant i\leqslant I}$ be a dataset of I images in $\mathcal{H}$. For every $i \in \{1, \ldots, I\}$, we build a noisy observation of $\overline{x}_i$, given by $y_i = \overline{x}_i + \sigma w_i$, where $\sigma > 0$ and $w_i \in \mathcal{H}$ is a realization of a standard normal i.i.d. random variable. To ensure the firm nonexpansiveness of the trained NN $\widetilde{J}$, we consider a regularized loss, incorporating an appropriate Lipschitz constraint on $Q_\theta = (2\widetilde{J}_\theta - \mathrm{Id})$ where $\theta \in \mathbb{R}^M$ are the learnable parameters. This constraint is applied to points $\widetilde{x}_i = \varrho_i \overline{x}_i + (1 - \varrho_i)\widetilde{J}(y_i)$, randomly selected on the segments $[\overline{x}_i, \widetilde{J}(y_i)]$, where ϱ_i is a realization of a random variable with uniform distribution on [0,1]. The training loss then reads

$$\Phi_i(\theta) = \|\widetilde{J}_\theta(y_i) - \overline{x}_i\|^2 + \lambda \max\left\{\|\boldsymbol{\nabla} Q_\theta(\widetilde{x}_i)\|_{\mathrm{S}}^2, 1 - \varepsilon\right\}, \tag{7}$$

where $\lambda > 0$ and $\varepsilon > 0$ are fixed parameters, $\|\cdot\|_{\mathrm{S}}$ is the spectral norm, and the regularization on the Jacobian aims at enforcing the nonexpansiveness of Q_θ. Training the network then amounts to minimize $\sum_{i=1}^{I} \Phi_i(\theta)$, which is differentiable and can thus be minimized using standard training algorithms such as, e.g., Adam [11].

2.3. Postprocessing network

In order to improve the reconstruction results, we propose to introduce a postprocessing step carried out by a specifically trained network G for the problem of interest. Precisely,

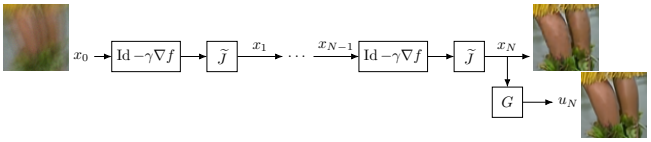

Fig. 1. Illustration of the proposed algorithm, relying on a firmly nonexpansive denoising network $\widetilde{J}$, and on a restoration network G.

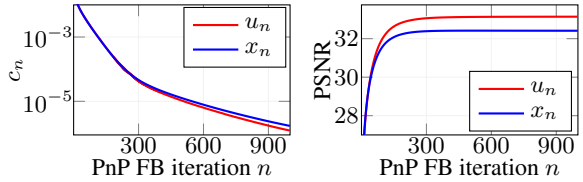

Fig. 2. Impact of the postprocessing along the iterations of Algorithm 8 on the BSD68 sample from Fig. 3.

defining $\widehat{x}$ the limit of $(x_n)_{n\in\mathbb{N}}$, we apply the network G as

$$\widehat{u} = G(\widehat{x}). \tag{8}$$

The proposed algorithm (4)-(8) is summarized in Fig. 1. Motivated by the success of UNets in a variety of image reconstruction and image-to-image translation tasks, we adopt the residual UNet architecture, similarly to [19].

3. EXPERIMENTS

3.1. Simulation setting

We apply our method to a deconvolution problem, which is an instance of (1) where $\mathcal{H} = \mathbb{R}^K$, and H models a blurring operator implemented as a circular convolution with impulse response h, and $e \in \mathbb{R}^K$ is a realization of an additive white Gaussian random noise with standard deviation $\nu > 0$.

In this work, we consider the same four problem settings presented in [20]: the Motion A (M. A) and Motion B (M. B) setups, i.e. h is the eighth and third motion test kernels respectively from [21] with $\nu = 0.01$, the Gaussian A (G. A) setup, i.e., h is a Gaussian kernel with standard deviation of 1.6 pixel and $\nu = 0.008$, and the Square (S.) setup, i.e., h is a square kernel of width of 7 pixels and $\nu = 0.01$.

3.2. Dataset and training

We choose the same DnCNN-based architecture as in [16] for $\widetilde{J}$ and we train it following the procedure described in Section 2.2. Precisely, we set $(\lambda, \varepsilon) = (10^{-5}, 5 \times 10^{-2})$ in (7) and we optimize the weights of the network $\widetilde{J}$ using Adam on pairs of groundtruth/noisy patches of size 50×50 (with $\sigma = 0.007$) built on the $50,000$ test images from ImageNet.

Regarding the training of the postprocessing UNet G, we create a dataset consisting of 6700 images with 200 images from the BSD training set [22], 1000 images from the COCO training set [23], 4500 images from the Waterloo dataset [24], 1000 images from the Div2K dataset [25]. For each image of our training set and in each of the 4 setups described above, we generate degraded measurements as in (1). We then solve each problem with 1000 iterations of the PnP-FB algorithm (4), thus obtaining a dataset of estimates $(\widehat{x}_i)_{1\leqslant i\leqslant I}$. The network G is trained by minimizing the ℓ_1-loss between the pairs $(G(\widehat{x}_i), \overline{x}_i)_{1\leqslant i\leqslant I}$ with the Adam algorithm for 8×10^5 iterations on randomly cropped patches of size 128×128, with a batch size of 16. The learning rate is set to 10^{-4} and is divided by 2 every 10^5 iterations. Eventually, we test our method on the 256×256 centered-crop Flickr30 and BSD500 datasets from [20], and both BSD68 and Kodak24 datasets [26].

3.3. Experimental results

For the sake of analyzing the interest of the postprocessing as a function of the number of iteration, we study the reconstruction results when one applies G at each iteration with $u_n = G(x_n)$. Since sequence $(x_n)_{n\in\mathbb{N}}$ converges to $\widehat{x}$, $(u_n)_{n\in\mathbb{N}}$ converges to $\widehat{u} = G(\widehat{x})$ if G is a continuous mapping. In Fig. 2, we show the behavior of Algorithm 8 on one image of the BSD68 dataset. The graph on the left shows the values of $c_n = \|x_n - x_{n-1}\|/\|x_0\|$ for the two sequences $(x_n)_{n\in\mathbb{N}}$ and $(u_n)_{n\in\mathbb{N}}$ and confirms the convergence of both sequences. The graph on the right shows that the positive effect of the postprocessing $u_n = G(x_n)$ can be observed from iteration 200, i.e. long before the number of iteration used for training of G.

We compare our method with different state-of-the-art methods. First, we consider pure optimization methods with either a TV prior (VAR method from [20]) or a constrained ℓ_1-minimization problem inducing sparsity in a redundant wavelet dictionary, inspired from [27]. Second, we compare our results with the unfolded network iRestNet [20]. Third, we consider PnP algorithms IRCNN [10], and DPIR [19], as well as a PnP-BM3D algorithm (i.e. algorithm (4) where $\widetilde{J}$ is replaced by BM3D [28]). We also compare with LMMO [16], that is the convergent PnP-FB algorithm (4) without the postprocessing step.

Table 1 gives average metrics on the Flickr30 and BSD500 test sets for the 4 different setups described in Section 3.1. Our method significantly improves over all methods, and performs on par with DPIR. Notice that our method tends to perform better than DPIR for more difficult Gaussian (G. A) and Square (S.) kernels, and sligthly worse than DPIR for the simpler motion blurs M. A and M. B. We can also notice the strong improvement over the LMMO method, i.e. our method without postprocessing, underlining the importance of the postprocessing step in the reconstruction quality.

Further comparisons are given in Table 2 on the BSD68 and Kodak24 test sets. On these datasets, we notice the lower performance of our method in terms of PSNR metric in the G. A and S. setups, while it continues to deliver state-of-the-art SSIM values. We believe that this might be due to a dif-

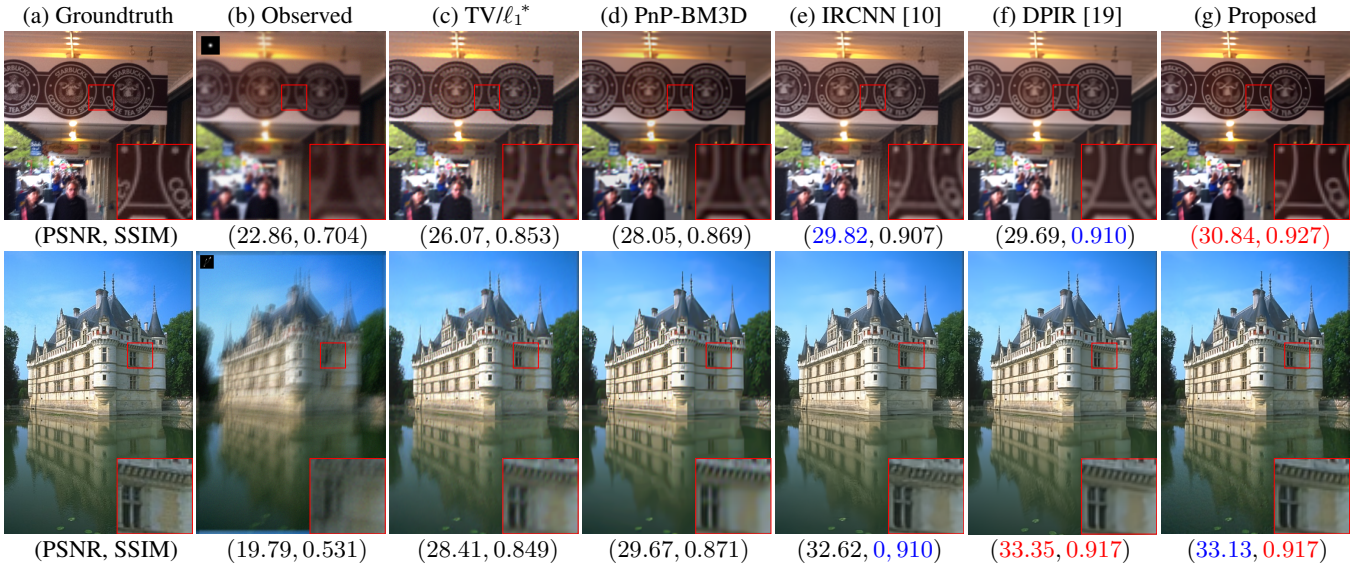

Fig. 3. Reconstructions on a Flickr30 sample (top row) in the G. A setup, and on a BSD68 sample (bottom row) in the M. A setup, for different methods. Best metrics are indicated in red, second best in blue and the blur kernel is shown in the top left corner of each observed image. *In column (c), the top row gives the TV solution and the second row gives the ℓ_1 solution. The TV result is borrowed from [20].

		TV	PnP-BM3D	iRestNet [20]	LMMO [16]	IRCNN [10]	DPIR [19]	Proposed
Flickr30	G.A	27.77/0.857	27.71/0.852	27.99/0.892	29.23/0.886	29.36/0.885	29.16/0.893	30.00/0.899
	M.A	28.22/0.856	29.91/0.885	30.49/0.919	32.18/0.923	32.50/0.927	33.40/0.938	33.14/0.937
	M.B	28.74/0.869	30.11/0.887	30.50/0.930	32.46/0.928	32.81/0.930	33.69/0.941	33.52/0.939
	S.	26.99/0.818	27.62/0.832	28.36/0.886	29.66/0.883	29.86/0.886	30.33/0.898	30.43/0.899
BSD500	G.A	27.30/0.804	27.02/0.785	27.94/0.853	28.43/0.833	28.66/0.840	28.45/0.844	28.73/0.854
	M.A	28.02/0.819	28.88/0.828	30.48/0.898	31.19/0.895	31.65/0.902	32.36/0.912	32.12/0.909
	M.B	28.41/0.829	29.01/0.827	30.83/0.910	31.46/0.900	31.96/0.907	32.58/0.916	32.51/0.915
	S.	26.38/0.756	26.53/0.749	27.90/0.840	28.40/0.822	28.74/0.834	28.90/0.837	29.04/0.845

Table 1. Average metrics on the Flickr30 and BSD500 datasets for the 4 different setups considered. Best values are indicated in red, second best in blue. Metrics for the TV prior and iRestNet are borrowed from [20].

		ℓ_1	IRCNN [10]	DPIR [19]	Proposed
BSD68	G.A	27.33/0.776	29.32/0.841	29.14/0.844	28.92/0.856
	M.A	28.38/0.814	32.04/0.901	32.67/0.910	32.39/0.909
	M.B	28.82/0.825	32.40/0.907	32.95/0.914	32.83/0.914
	S.	26.71/0.745	29.26/0.832	29.40/0.833	29.38/0.845
Kodak24	G.A	28.03/0.789	30.36/0.851	30.02/0.856	28.54/0.863
	M.A	28.92/0.810	33.14/0.897	33.88/0.907	33.55/0.904
	M.B	29.67/0.826	33.62/0.904	34.28/0.914	34.06/0.912
	S.	28.05/0.768	30.67/0.846	30.92/0.850	30.01/0.857

Table 2. Average metrics on the BSD68 and Kodak24 datasets for the 4 different setups considered. Best values are indicated in red, second best in blue.

ficulty of our UNet to generalize on images much larger than the 128×128 patches on which it was trained.

Fig. 3 shows a visual reconstruction results for a G. A setup on a sample from Flickr30 (top row) and M. A setup on a sample from BSD68 (bottom row). We observe that our method reconstructs fine details and shows more features than reconstructions by other methods. This is visible for instance at the bottom of the BSD68 sample, where the original image shows a trembling reflection of the castle in the water: our method leads to the nicer visual estimate for this effect.

4. CONCLUSION

In this paper, we have proposed an approach for building convergent PnP algorithms. This is done by leveraging maximally monotone operator theory, and constraining the Lipschitz constant of the denoiser during the training process. We proposed to pair the resulting PnP-FB algorithm with a post-processing network trained on the problem of interest. We applied our method on several image deconvolution problems, showing that it leads to outstanding restoration quality, comparable with the most advanced state-of-the-art methods.

5. REFERENCES

[1] Patrick L Combettes and Jean-Christophe Pesquet, "Proximal splitting methods in signal process.," in *Fixed-point algorithms for inverse problems in science and engineering*, pp. 185–212. Springer, 2011.

[2] George HG Chen and R Tyrrell Rockafellar, "Convergence rates in forward–backward splitting," *SIAM J. Optim.*, vol. 7, no. 2, pp. 421–444, 1997.

[3] Patrick L Combettes and Valérie R Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.*, vol. 4, no. 4, pp. 1168–1200, 2005.

[4] Heinz H Bauschke and Patrick L Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, Springer, 2017.

[5] Leonid I Rudin, Stanley Osher, and Emad Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, no. 1-4, pp. 259–268, 1992.

[6] Kristian Bredies, Karl Kunisch, and Thomas Pock, "Total generalized variation," *SIAM J. on Imaging Sci.*, vol. 3, no. 3, pp. 492–526, 2010.

[7] Laurent Condat, "Semi-local total variation for regularization of inverse problems," in *Proc. of the IEEE Eur. Signal Process. Conf.*, 2014, pp. 1806–1810.

[8] Ingrid Daubechies, Michel Defrise, and Christine De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math.*, vol. 57, pp. 1413–1457, 2004.

[9] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg, "Plug-and-play priors for model based reconstruction," in *IEEE Global Conf. on Signal and Information Process.*, 2013, pp. 945–948.

[10] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recognit.*, 2017, pp. 3929–3938.

[11] Kai Zhang, Wangmeng Zuo, and Lei Zhang, "Deep plug-and-play super-resolution for arbitrary blur kernels," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recognit.*, 2019, pp. 1671–1681.

[12] Stanley H Chan, Xiran Wang, and Omar A Elgendy, "Plug-and-play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Trans. on Comput. Imaging*, vol. 3, no. 1, pp. 84–98, 2016.

[13] Ernest Ryu, Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin, "Plug-and-play methods provably converge with properly trained denoisers," in *Int. Conf. on Mach. Learn.*, 2019, vol. 97, pp. 5546–5557.

[14] Matthieu Terris, Audrey Repetti, Jean-Christophe Pesquet, and Yves Wiaux, "Building firmly nonexpansive convolutional neural networks," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Process.*, 2020, pp. 8658–8662.

[15] Johannes Hertrich, Sebastian Neumayer, and Gabriele Steidl, "Convolutional proximal neural networks and plug-and-play algorithms," *arXiv preprint arXiv:2011.02281*, 2020.

[16] Jean-Christophe Pesquet, Audrey Repetti, Matthieu Terris, and Yves Wiaux, "Learning maximally monotone operators for image recovery," *to appear in SIAM J. on Imaging Sci.*, 2021.

[17] Tim Meinhardt, Michael Moller, Caner Hazirbas, and Daniel Cremers, "Learning proximal operators: Using denoising networks for regularizing inverse imaging problems," in *Proc. of the IEEE Int. Conf. on Comput. Vision*, 2017, pp. 1781–1790.

[18] Kaixuan Wei, Angelica Aviles-Rivero, Jingwei Liang, Ying Fu, Carola-Bibiane Schönlieb, and Hua Huang, "Tuning-free plug-and-play proximal algorithm for inverse imaging problems," in *Int. Conf. on Mach. Learn.*, 2020, vol. 119, pp. 10158–10169.

[19] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte, "Plug-and-play image restoration with deep denoiser prior," *IEEE Trans. on Pattern Anal. and Mach. Intelligence*, 2021.

[20] Carla Bertocchi, Emilie Chouzenoux, Marie-Caroline Corbineau, Jean-Christophe Pesquet, and Marco Prato, "Deep unfolding of a proximal interior point method for image restoration," *Inverse Problems*, vol. 36, no. 3, pp. 034005, 2020.

[21] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman, "Understanding and evaluating blind deconvolution algorithms," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recognit.*, 2009, pp. 1964–1971.

[22] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. of the IEEE Int. Conf. on Comput. Vision*, 2001, vol. 2, pp. 416–423.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft COCO: common objects in context," in *Eur. Conf. on Comput. Vision*. Springer, 2014, pp. 740–755.

[24] Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang, "Waterloo Exploration Database: New challenges for image quality assessment models," *IEEE Trans. on Image Process.*, vol. 26, no. 2, pp. 1004–1016, Feb. 2017.

[25] Eirikur Agustsson and Radu Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recognit. Workshops*, 2017, pp. 126–135.

[26] Rich Franzen, "Kodak lossless true color image suite," 1999, [Online], Available: http://r0k.us/graphics/kodak.

[27] Rafael E Carrillo, Jason D McEwen, Dimitri Van De Ville, Jean-Philippe Thiran, and Yves Wiaux, "Sparsity averaging for compressive imaging," *IEEE Signal Process. Lett.*, vol. 20, no. 6, pp. 591–594, 2013.

[28] Ymir Mäkinen, Lucio Azzari, and Alessandro Foi, "Collaborative filtering of correlated noise: Exact transform-domain variance for improved shrinkage and patch matching," *IEEE Trans. Image Process.*, vol. 29, pp. 8339–8354, 2020.